Adaptive Direct FEM Simulation with Unicorn/FEniCS-HPC for CS1

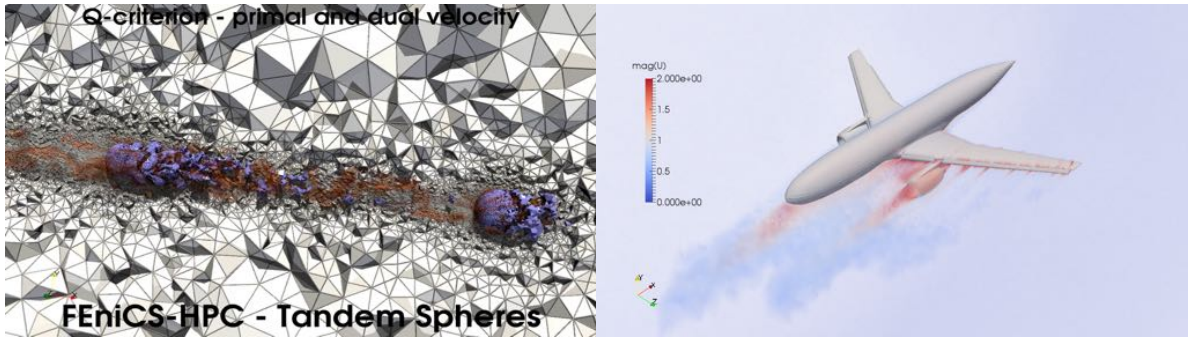
Johan Jansson[12], Ezhilmathi Krishnasamy[2], Massimiliano Leoni[12]

Collaborators: Cem Degirmenci, Laura Saavedra, Johan Hoffman, Niclas Jansson, Philipp Schlatter, Ricardo Vinuesa, Jing Gong, Peter Vincent

Computational Science and Technology CSC, KTH, Stockholm [1]

BCAM - Basque Center for Applied Mathematics, Bilbao [2]

5th International Workshop on High Order CFD Methods
Kissimmee, FL, January 6-7, 2018



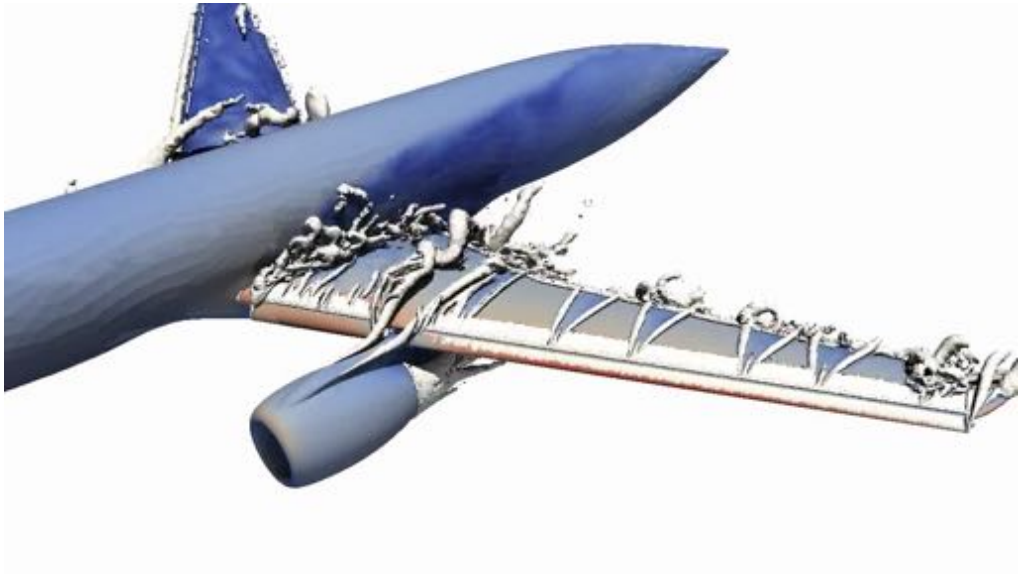
Aims and outline

Adaptive Direct FEM Simulation (DFS) results with P1 for CS1 together with comparison to workshop-provided meshes.

Preliminary comparison to other leading high-order open source packages: PyFR and Nek5000

Draft paper available on my home page: `http://www.csc.kth.se/~jjan`

Part of broader program for predictive adaptive DFS high and low Re simulation for industrial challenge problems.



[stall prediction in HiLiftPW3]

Automated Computational Modeling

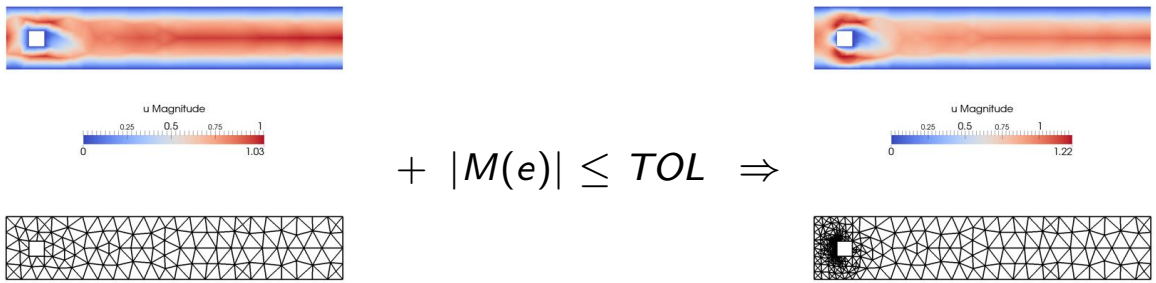
FEniCS(-HPC) open source FEM framework for automated solution of general PDE and Direct FEM Simulation (DFS) methodology enables:

Automated discretization (generate code for linear system from PDE):

```
r = (inner(grad(u), grad(v)) - inner(f, v))*dx
```

$\Rightarrow$ `Poisson.cpp`

Automated error control (incl. parallel adaptive mesh refinement):



$+ \ |M(e)| \leq TOL \ \Rightarrow$

with $M(e)$ a goal functional of the computational error $e = u - U$.

Automated modeling of unresolved subscales (i.e. turbulence):

$(R(U), v) + (hR(U), R(v)) = 0, \forall v \in V_h$ (ILES: residual-based stabilization, Adaptive DNS/LES)

Goal: Autom. generate the solution, mesh and program from PDE (residual) and goal functional $M(e)$ (e.g. drag).

Direct FEM Simulation (DFS) methodology

Developed over a 20+ year period by Johnson, Hoffman, Jansson, etc.

Realized in the open source Unicorn/FEniCS-HPC software framework (available on BitBucket).
Incompressible Navier-Stokes in strong residual form:

$$R(\hat{u}) \quad = \quad \begin{cases} \partial_t u + (u \cdot \nabla)u - \nu \Delta u + \nabla p = 0 \\ \nabla \cdot u = 0 \end{cases} \qquad \hat{u} = (u, p)$$

Weak residual $r(\hat{u}, \hat{v}) = (R(\hat{u}), \hat{v})$
No explicit turbulence model

Space-time cG(1)cG(1) FEM with Galerkin/least squares stabilization (acting as ILES)

$$\bar{r}(\hat{U}, \hat{v}) = r(\hat{u}, \hat{v}) + (\delta R(\hat{U}), R(\hat{v})) = 0$$

$$\delta = \kappa min(h), \ \kappa \approx 1, \ \forall \hat{v} \in \hat{V}_h, \ \hat{U} \in \hat{V}_h$$

Do-nothing duality-based adaptive error control method

Directly use err repr. $M(\hat{e}) = r(\hat{U}, \hat{\phi})$ gives error indicator $\mathcal{E}_K = r(\hat{U}, \hat{\phi})_K$
Adjoint prob. autom. gen.: $r'(\hat{\phi}, \hat{v}) = M(\hat{v})$
Iteratively solve primal and dual problem, refine marked cells.

DFS acts as Adaptive DNS/LES

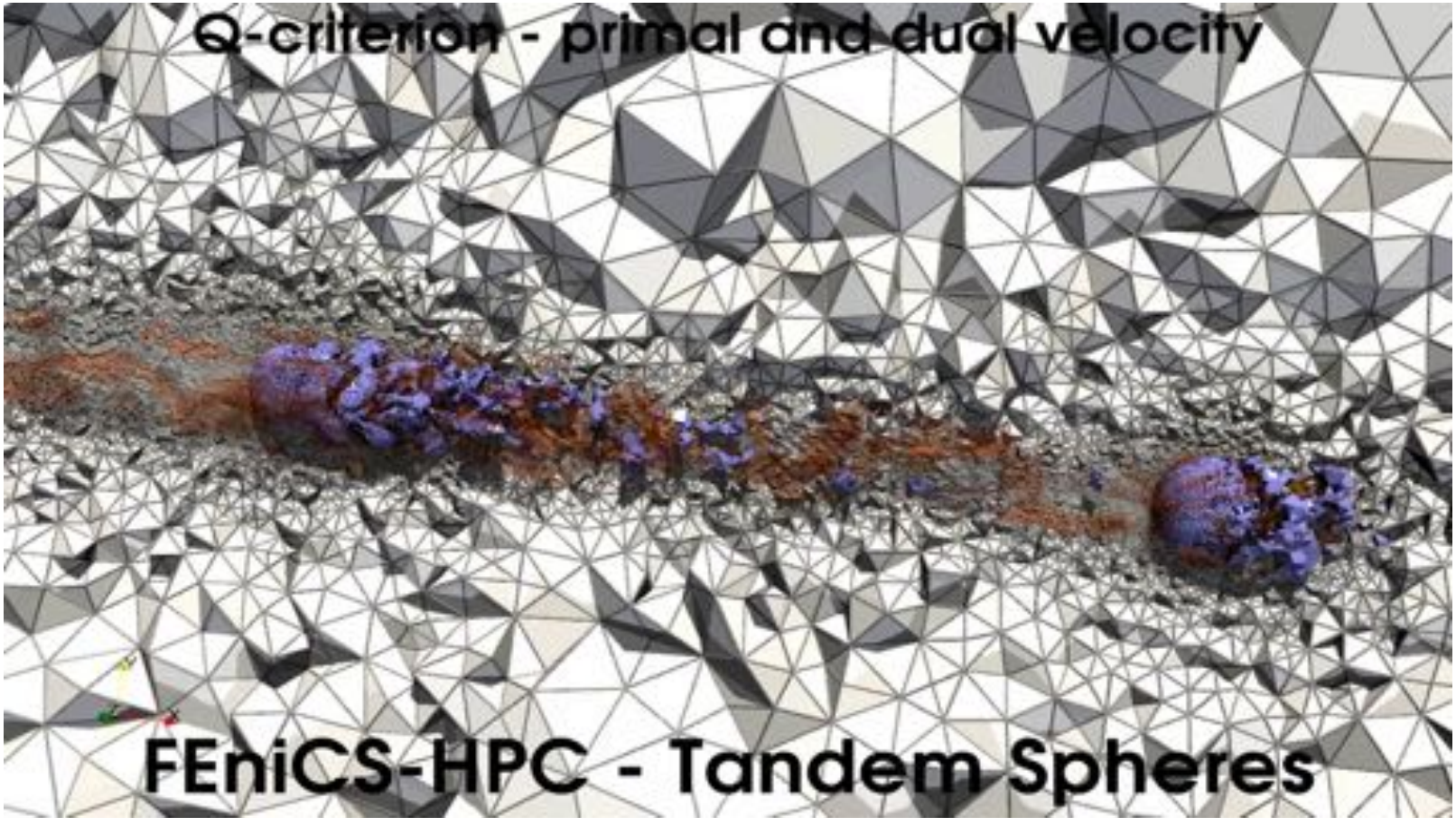
Large time steps by Schur-preconditioned fixed-point

Segregated fixed-point iteration between u and p, gives straightforward preconditioning of linear systems:
CG+bjacobi/ILU for continuity and BiCGStab+bjacobi/ILU (PETSc)
Schur-complement preconditioning allows large time-steps, up to 100x larger than simple fixed-point.

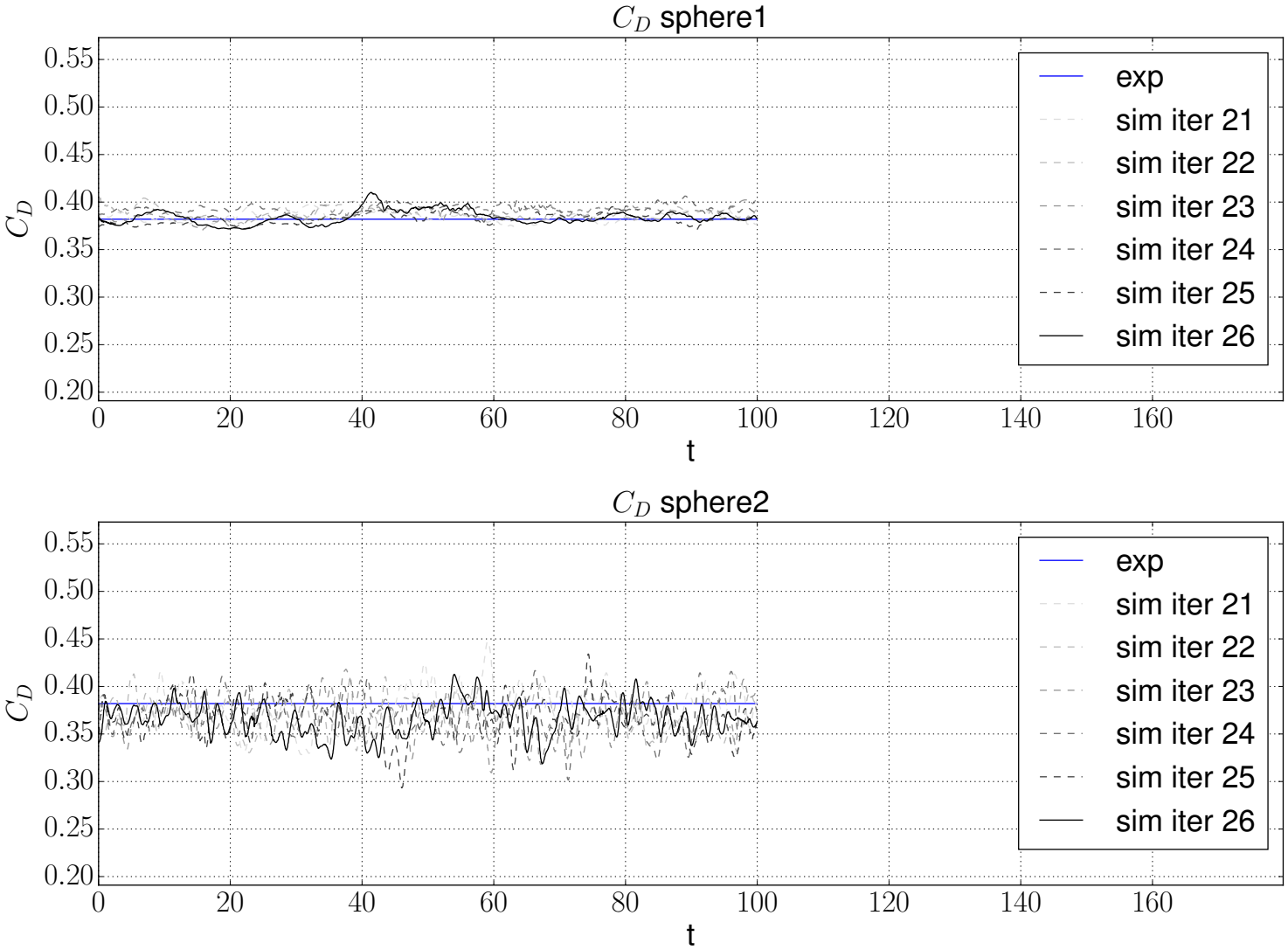
[Hoffman, Jansson, et. al., 2012 C&F], [Hoffman, Jansson, et. al. 2016, ECM], [Jansson, et. al. 2018, HiLift, Springer Brief]

CS1: Viz of adaptive results

Goal quantity: drag $M(\hat{u})$
Recall err. rep.: $M(\hat{e}) = (-R(\hat{U}), \hat{\phi})$
Adjoint $\hat{\phi}$ weights error indicators upstream, zero weight downstream of downstream sphere, residual $R(\hat{U})$ weights turbulent wakes.
See extra material for ref on computability by adjoint stability.



Time-evolution of C_D



Time evolution of the drag coefficient for various iterations of our adaptive procedure.

Convergence order

We hope to contribute an interesting perspective on convergence order in the setting of adaptive methods.

What is the order of convergence of an adaptive h-refinement method?

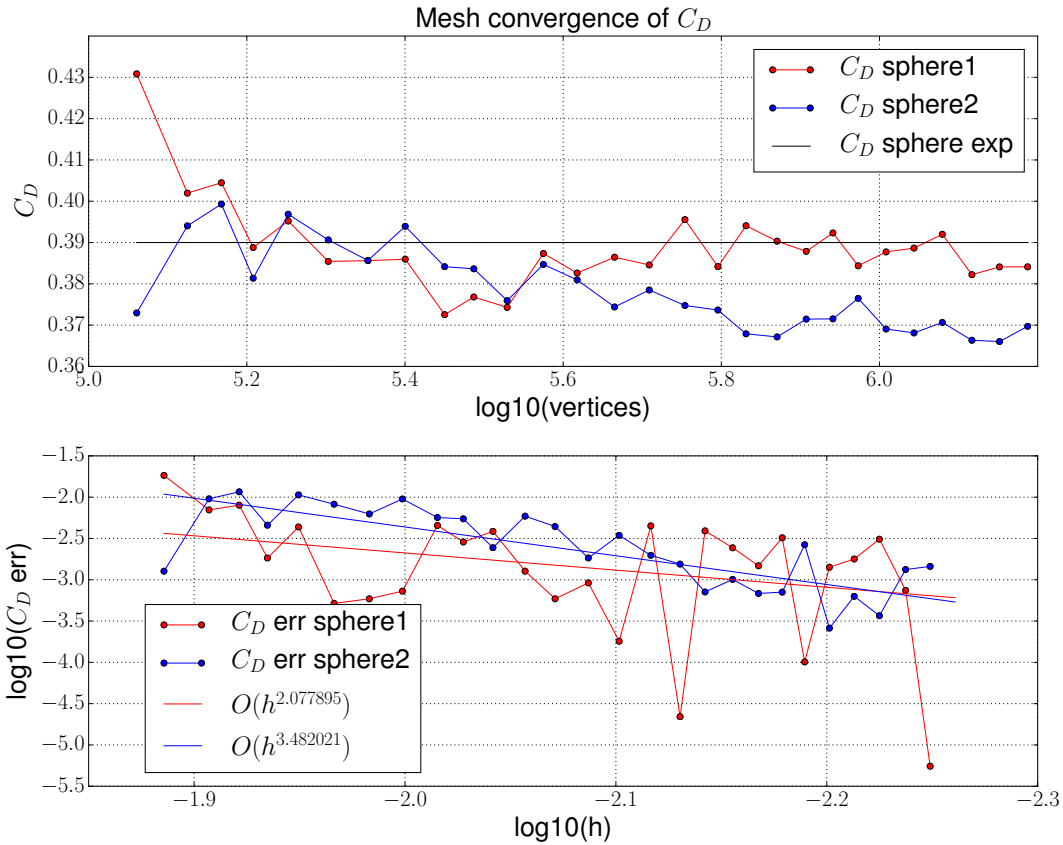
- Generalized length scale $h = N_{DOF}^{\frac{1}{d}}$
- Order of convergence $e(h) = Ch^p$ or $log(e(h)) = p\, log(Ch)$
- Compute convergence sequence (e_i, h_i)
- Least-squares fit for p gives "effective order of convergence".

Adaptive DNS/LES $\Rightarrow$ asymptotic regime is DNS

We are interested in efficient computation $\Rightarrow$ as coarse meshes as possible typically in ILES regime which is non-smooth (refinement uncovers finer scales).

Uniform refinement quasi-optimal for smooth solutions.

Adaptive mesh convergence

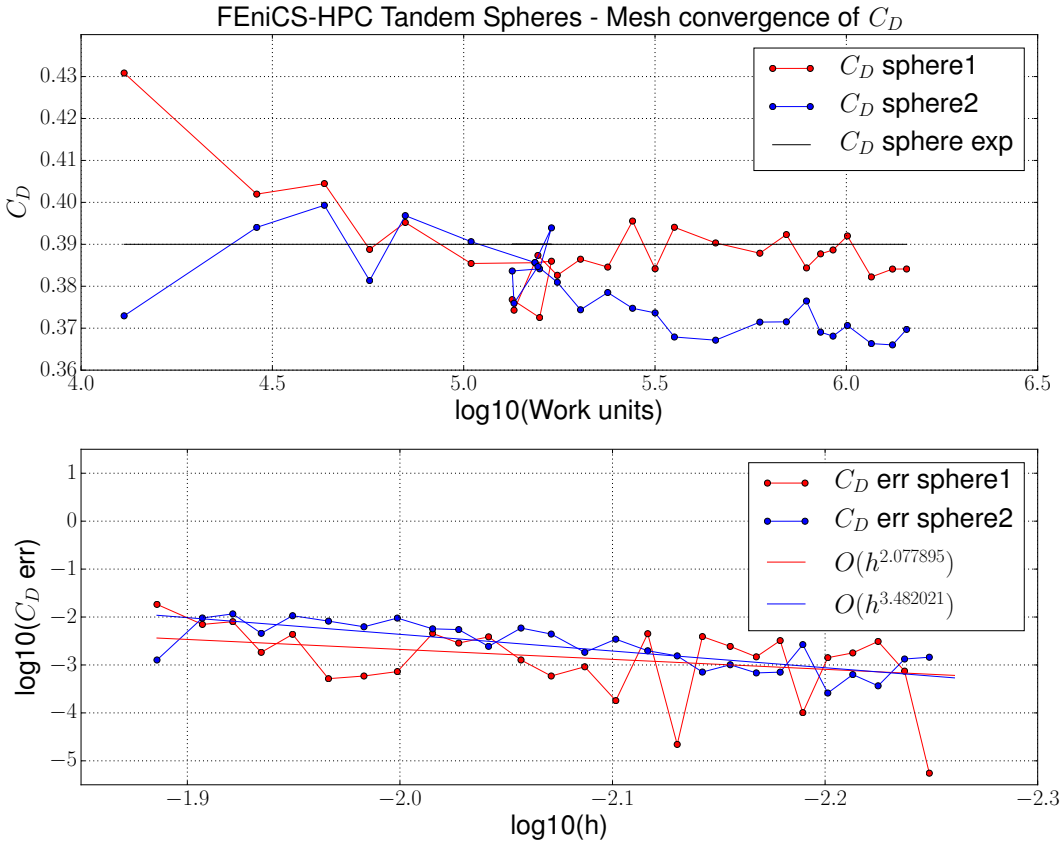


Mesh convergence of the drag coefficients of the two spheres. Effective order of convergence > 3. However, not interesting with asymptotic behavior. Allowing coarsening could give different behavior, (Couchman/Galbraith).

We see a clear drag reduction for the downstream sphere, consistent with a "slipstream" phenomenon.

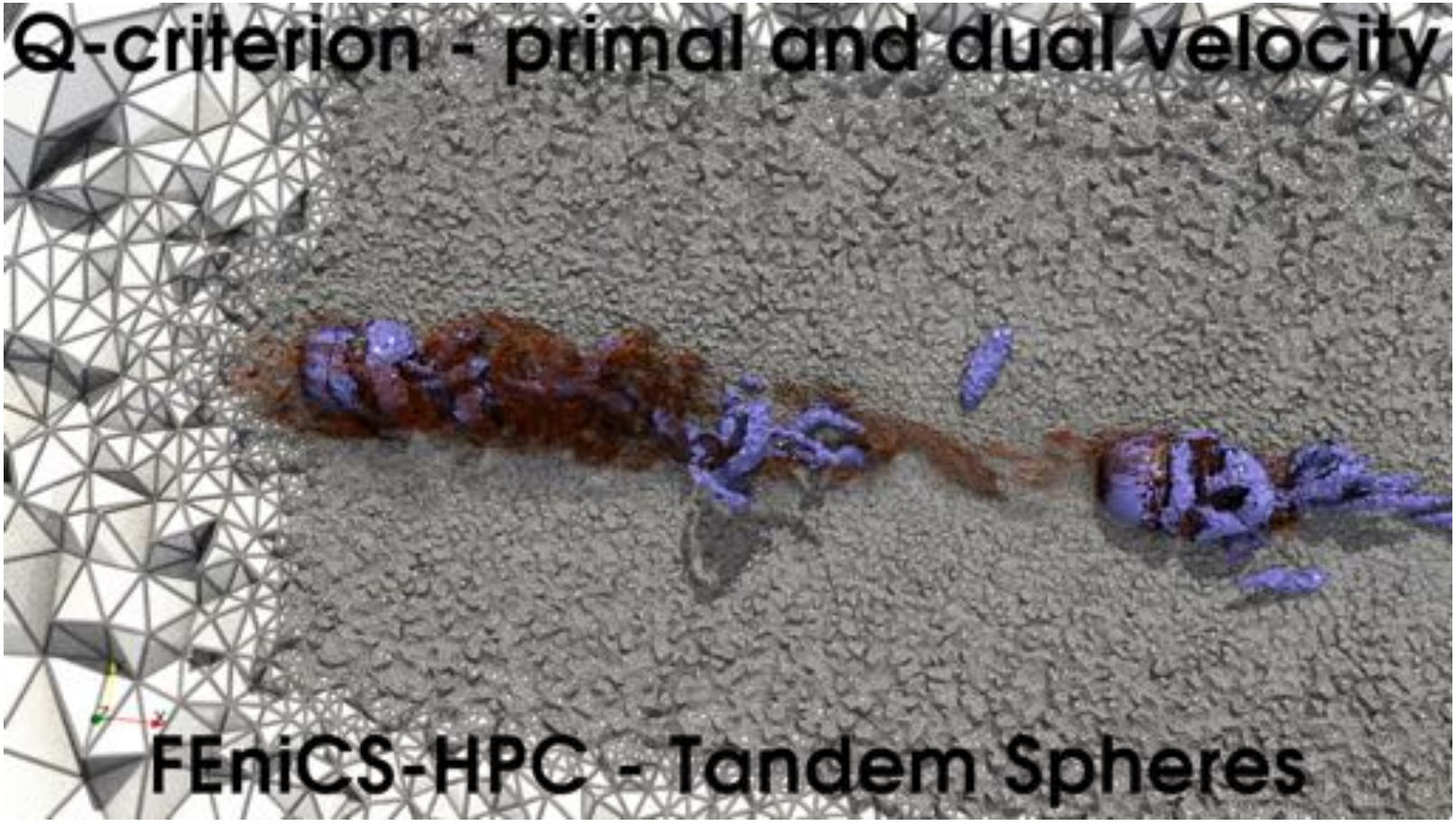
2% away from experimental reference for upstream sphere.

Adaptive mesh convergence



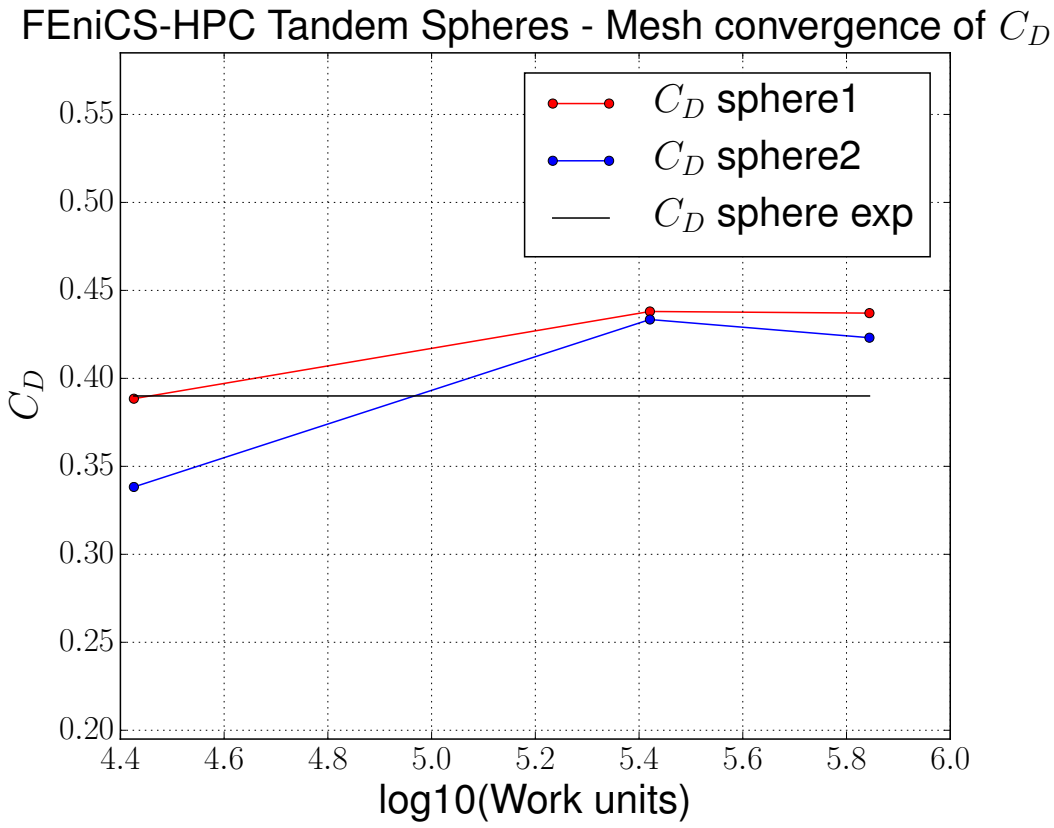
Mesh convergence of the drag coefficients of the two spheres.

CS1: Viz of provided Pointwise-Mesh5



10% away from experimental reference for upstream sphere, no clear drag reduction for downstream sphere.

CS1: Pointwise mesh convergence



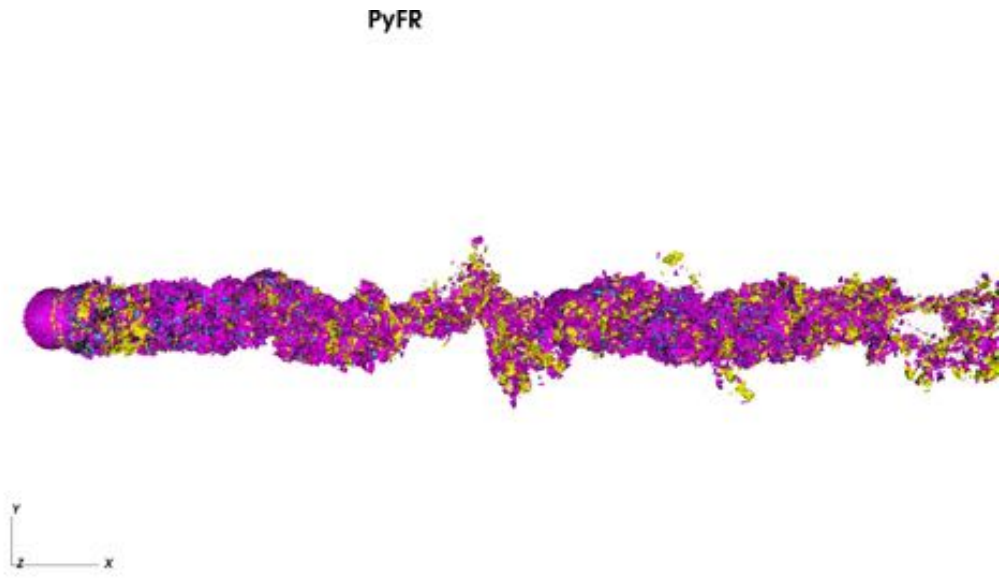
Mesh convergence of the drag coefficients of the two spheres, workshop meshes 3-5.

10% away from experimental reference for upstream sphere, no clear drag reduction for downstream sphere.

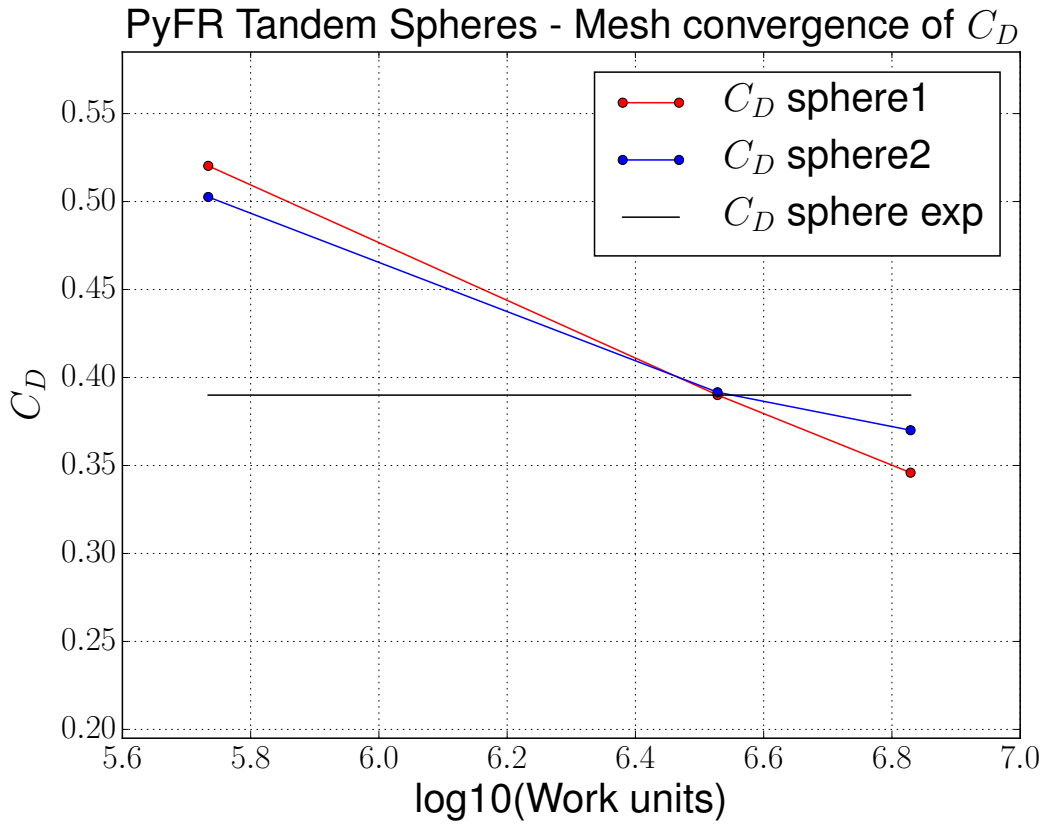
PyFR - Pointwise Mesh1P3

PyFR simulations by: Ezhilmathi Krishnasamy (KTH)

Main observation: severe timestep restriction $dt \approx 10^{-4}$ (3 orders of magnitude smaller than Unicorn/FEniCS-HPC). Implies many time steps and ~ 1 order of magnitude higher cost than Unicorn/FEniCS just for Mesh1P3). CPU simulations, possibility for 5x speedup with GPU (ref. conversation with Peter Vincent).



CS1: PyFR mesh convergence



Mesh convergence of the drag coefficients of the two spheres, PyFR Mesh1 P1-3.

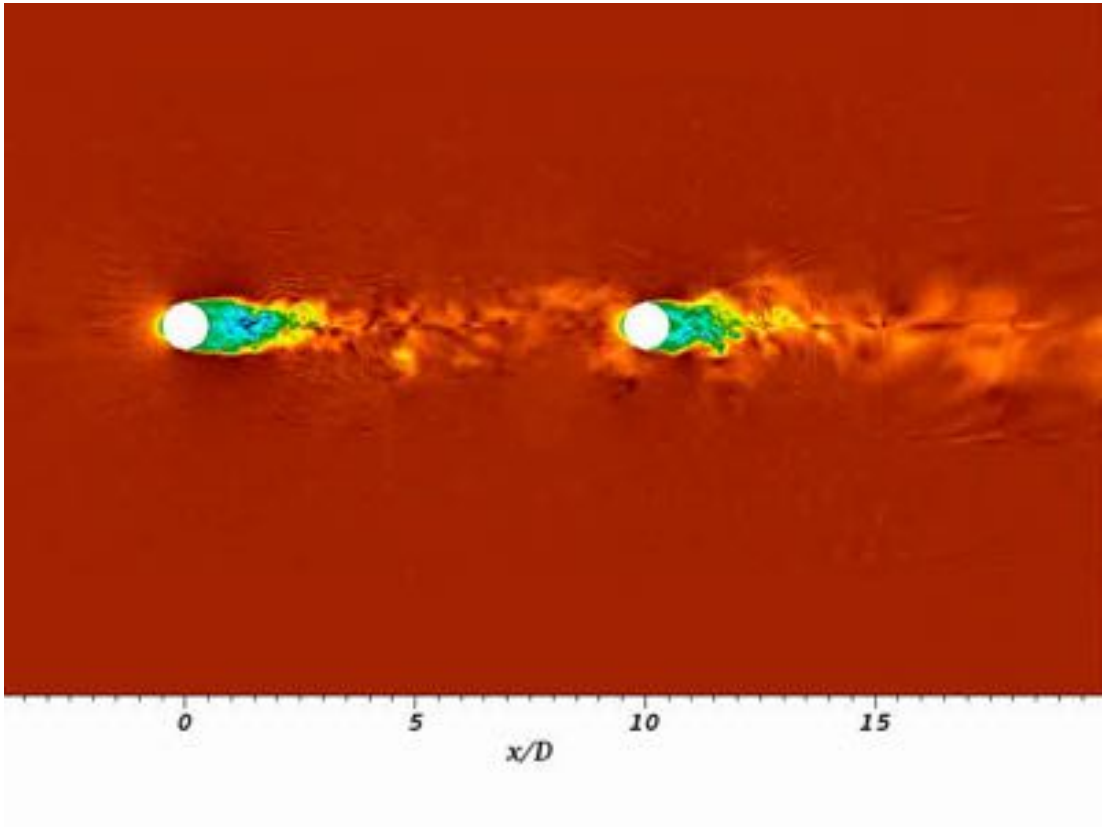
No clear mesh convergence, no clear drag reduction for downstream sphere. $\sim$ 1 order of magnitude higher cost than Unicorn/FEniCS just for Mesh1P3). Work in progress to compute on more meshes/orders.

Preliminary Nek5000 simulation

Nek5000 simulation by: Ricardo Vinuesa, Jing Gong and Philipp Schlatter (KTH)

ILES (or under-resolved DNS) regime with 1e7 grid points (4e7 DOFs) and $dt \approx 10^{-3}$

Work units: 5e6 Wall clock runtime: 13h C_D values in progress



MOOC - KTH/edX online course on adaptive FEM and FEniCS

High Performance Finite Element Modeling MOOC supported by KTH MOOC committee

Opened on October 17, 2017, already 4500+ students, join the fun!

Advanced part 2 on DFS, turbulence and HPC opens mid-March 2018.

Summary

- Adaptive DFS in Unicorn/FEniCS-HPC **captures relevant qualitative and quantitative results to 2% of exp.** Parameter-free, no explicit turb. modeling.
- **Adaptivity and large timesteps allow high efficiency and accuracy**: meshes of ca. 1M vertices, wall clock $\sim$1h using 1k cores.
- We wish to give perspective on convergence order in setting of adaptive methods. Effective order of convergence > 3 with cG(1) (P1).
- Open source pays off: Unicorn 2017-2018 $\sim$2 orders of magnitude faster than Unicorn 2012-2014 (developed in-house for a few years), mostly due to Schur formulation enabling large time-steps (ref. HiLiftPW2-3).
- Verification and validation of do-nothing duality-based adaptive method for low Re turbulent flow. Automated formulation, no jump terms necessary or manual formulation.

Preliminary comparisons to leading open source high-order packages:

- Accurate Unicorn/FEniCS-HPC simulation $\sim$1 order of magnitude cheaper+faster than preliminary Nek5000 simulation, drag comparison work in progress.
- Accurate Unicorn/FEniCS-HPC simulation $\sim$1 order of magnitude cheaper+faster than PyFR simulations for finest PyFR simulation, severe timestep restriction for PyFR.

Future work:

- Can an ensemble-approach to parallel-in-time significantly extend parallel strong scaling?
- Extension to compressible flow, initial work for model problems already done.
- Investigate r-adaptivity, allows coarsening.

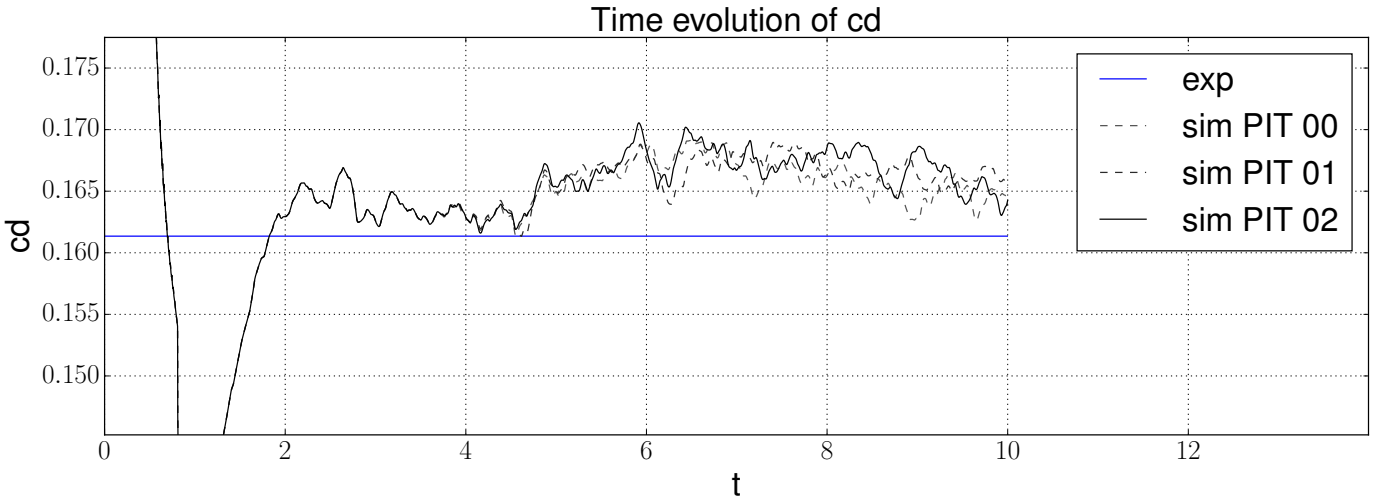
Acknowledgements:

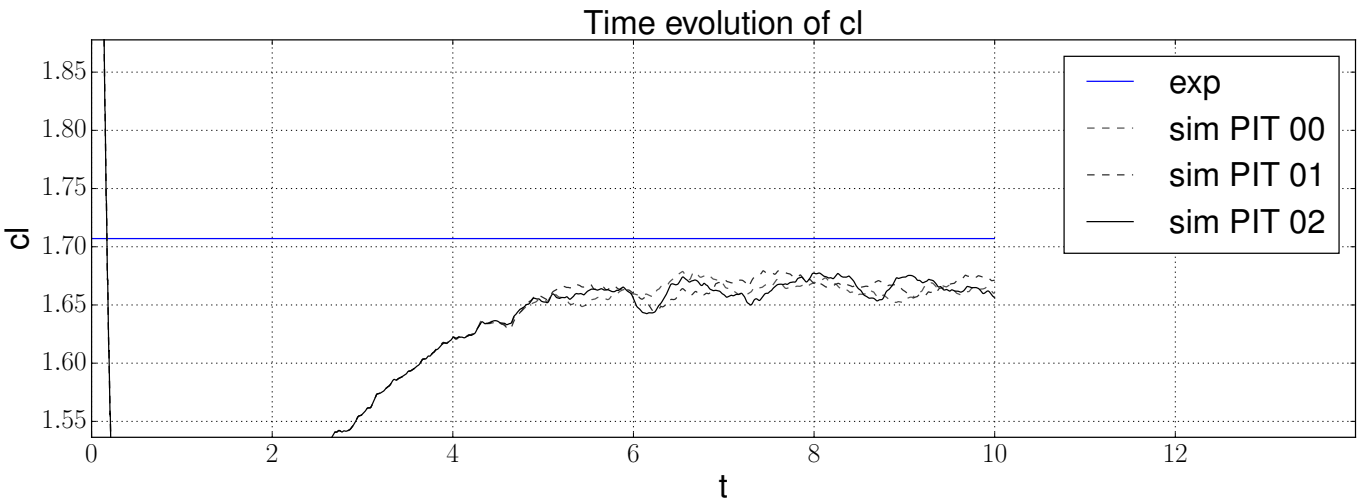
MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD
European Commission
EXCELENCIA SEVERO OCHOA
EUSKO JAURLARITZA GOBIERNO VASCO
SNIC

Extra material

Parallel-in-time

Ensemble approach of partitions of time interval for bluff body, here $\alpha = 4.36$





[Jansson, Hoffman, preprint, 2017]

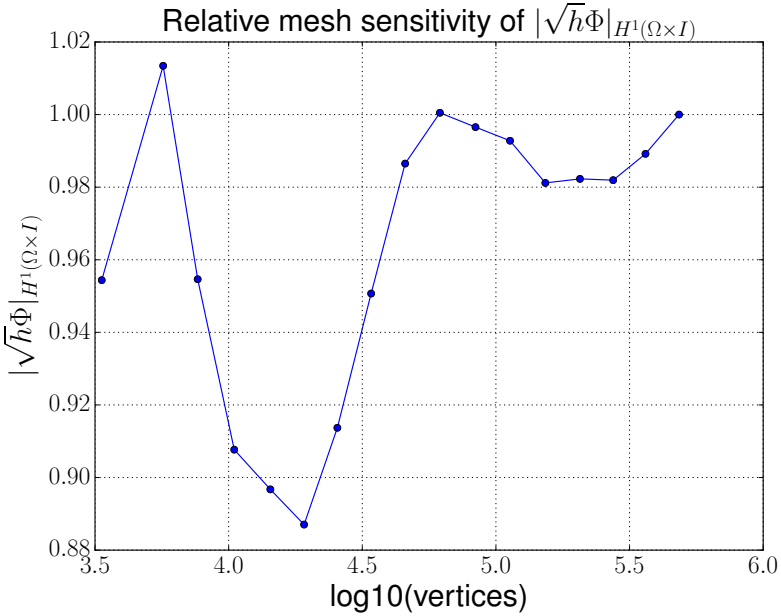
DFS computability of turbulent flow by adjoint stability

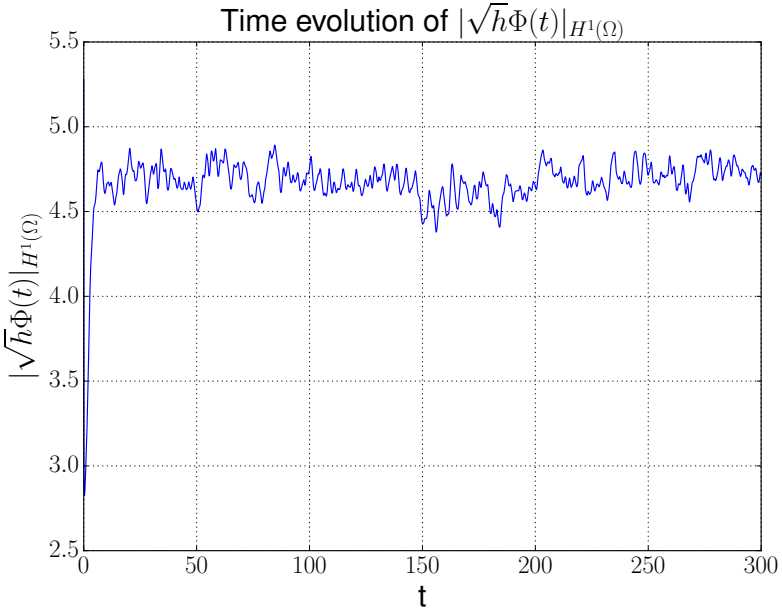
We demonstrate computability of 3D turbulent flow by stability of the 3D adjoint problem. We describe this in the invited chapter "Computability and Adaptivity in CFD" in Encyclopedia of Computational Mechanics.

We can bound the global error by:

$$|M(\hat{u}) - M(\hat{U})| \leq C_U h \|R(\hat{U})\| |\hat{\psi}|_{H^1} \tag{1}$$

We show that $\|\sqrt{h}R(\hat{U})\|$ and $|\sqrt{h}\hat{\psi}|_{H^1}$ are bounded, which gives computability.





Stability of H1-seminorm of adjoint velocity wrt. mesh refinement and time

MSO4SC

Mathematical Modelling, Simulation and Optimization for Societal Challenges with Scientific Computing

European Commission

Horizon 2020
European Union funding
for Research & Innovation

H2020 project with BCAM, KTH, Atos, EU-maths-in, etc.

Objective: Construct an e-infrastructure that provides simple access to supercomputing with open source mathematical frameworks such as FEniCS-HPC, develop and manage simulations through a web-interface with multiple infrastructures (supercomputers, cloud systems, etc.).

Enables simple, automatic and optimal management of ensemble and adaptive simulation on a heterogenous set of computer resources.

`mso4sc.eu`